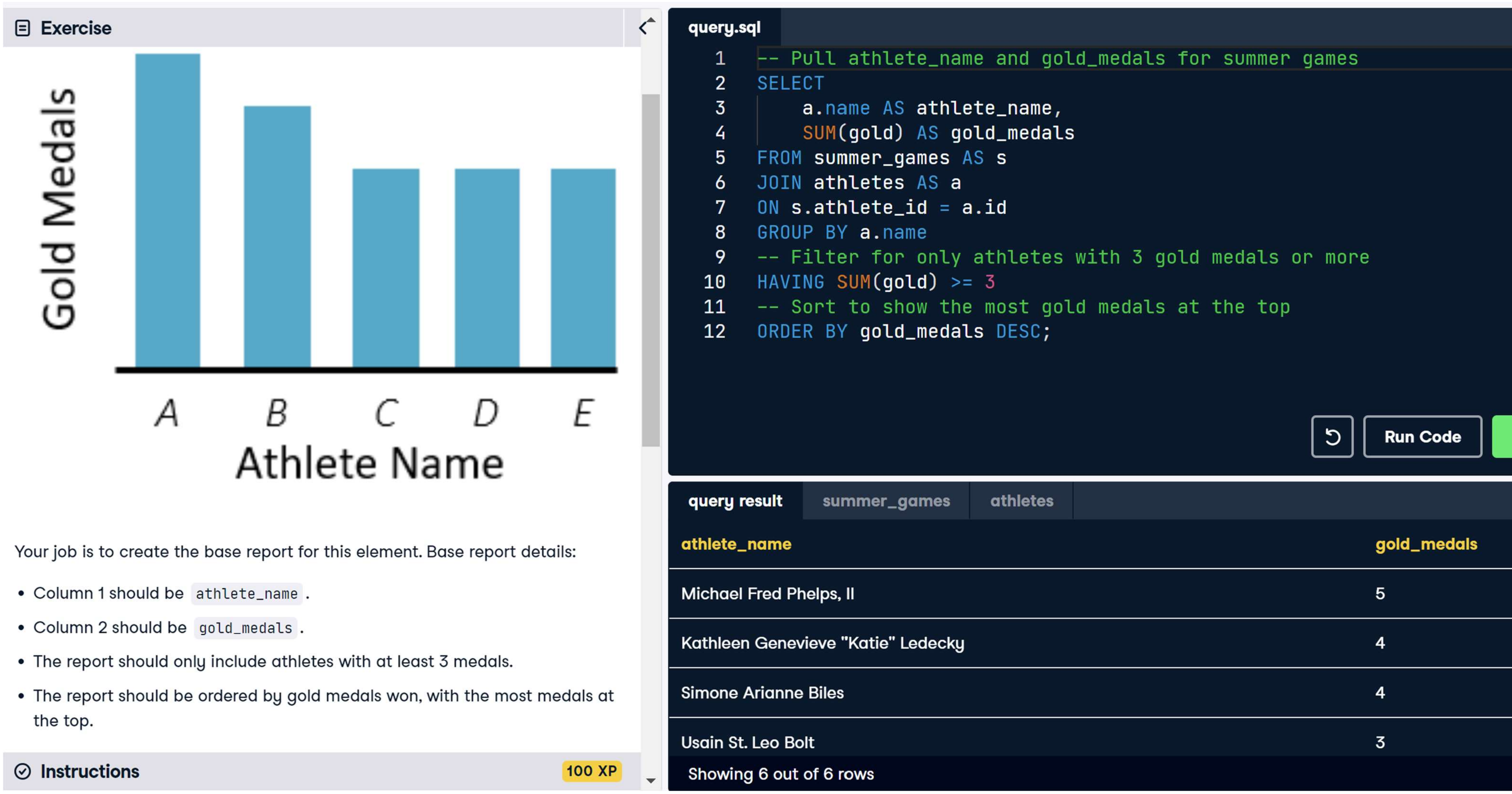
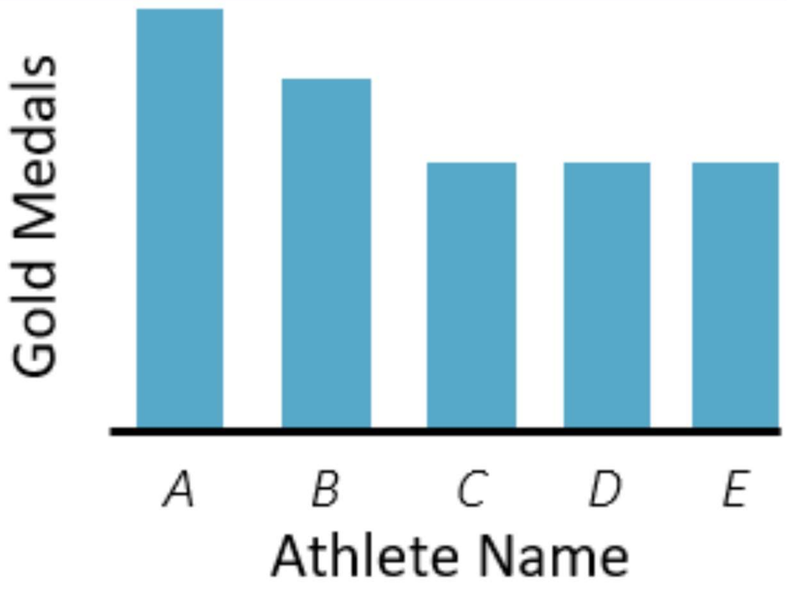




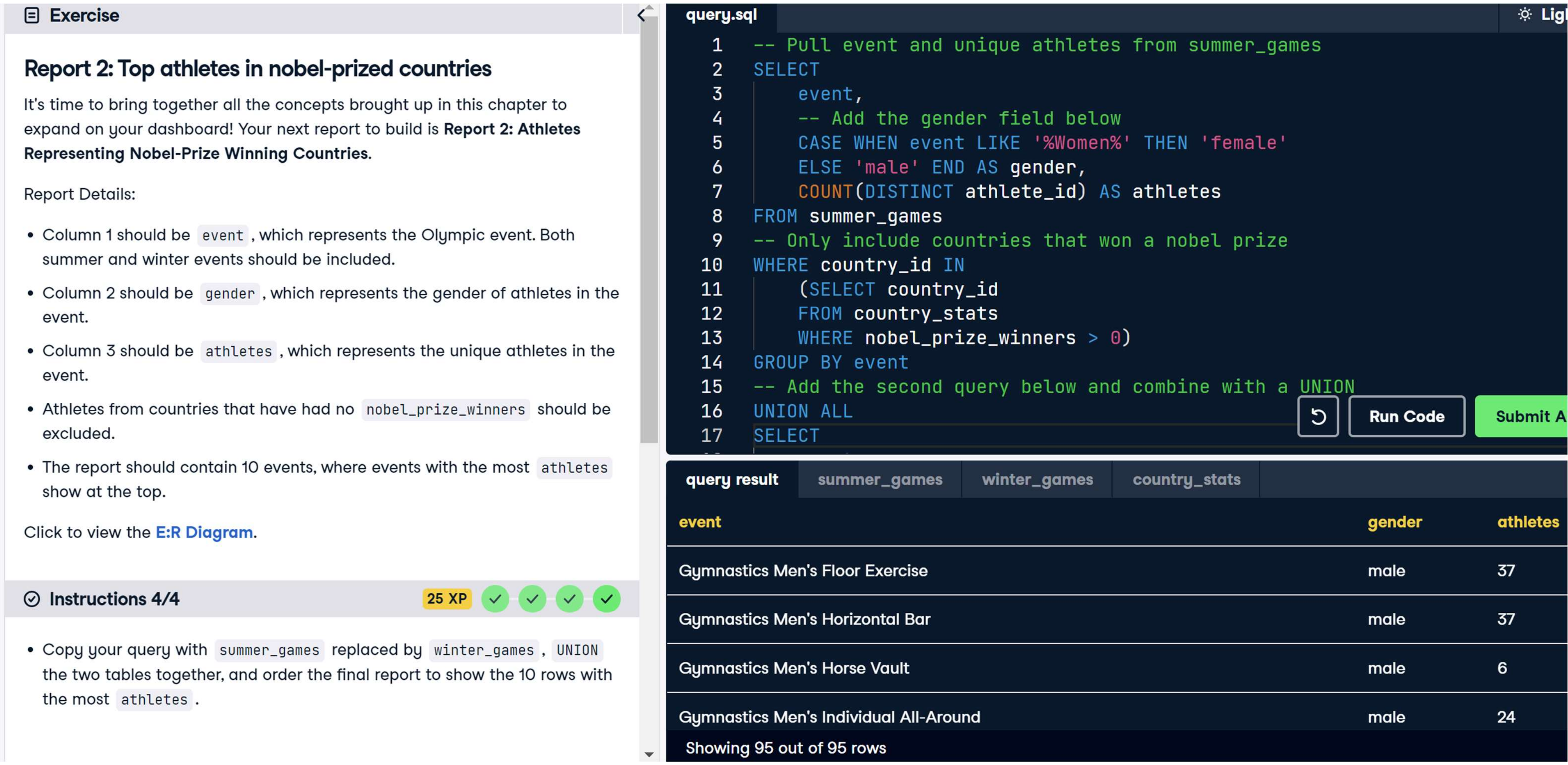
Your job is to create the base report for this element. Base report details:

- Column 1 should be `athlete_name`.
- Column 2 should be `gold_medals`.
- The report should only include athletes with at least 3 medals.
- The report should be ordered by gold medals won, with the most medals at the top.

```sql
-- Pull athlete_name and gold_medals for summer games
SELECT
    a.name AS athlete_name,
    SUM(gold) AS gold_medals
FROM summer_games AS s
JOIN athletes AS a
ON s.athlete_id = a.id
GROUP BY a.name
-- Filter for only athletes with 3 gold medals or more
HAVING SUM(gold) >= 3
-- Sort to show the most gold medals at the top
ORDER BY gold_medals DESC;
```

athlete_name	gold_medals
Michael Fred Phelps, II	5
Kathleen Genevieve "Katie" Ledecky	4
Simone Arianne Biles	4
Usain St. Leo Bolt	3

Showing 6 out of 6 rows



Exercise

Report 2: Top athletes in nobel-prized countries

It's time to bring together all the concepts brought up in this chapter to expand on your dashboard! Your next report to build is **Report 2: Athletes Representing Nobel-Prize Winning Countries.**

Report Details:

- Column 1 should be `event`, which represents the Olympic event. Both summer and winter events should be included.
- Column 2 should be `gender`, which represents the gender of athletes in the event.
- Column 3 should be `athletes`, which represents the unique athletes in the event.
- Athletes from countries that have had no `nobel_prize_winners` should be excluded.
- The report should contain 10 events, where events with the most `athletes` show at the top.

Click to view the E:R Diagram.

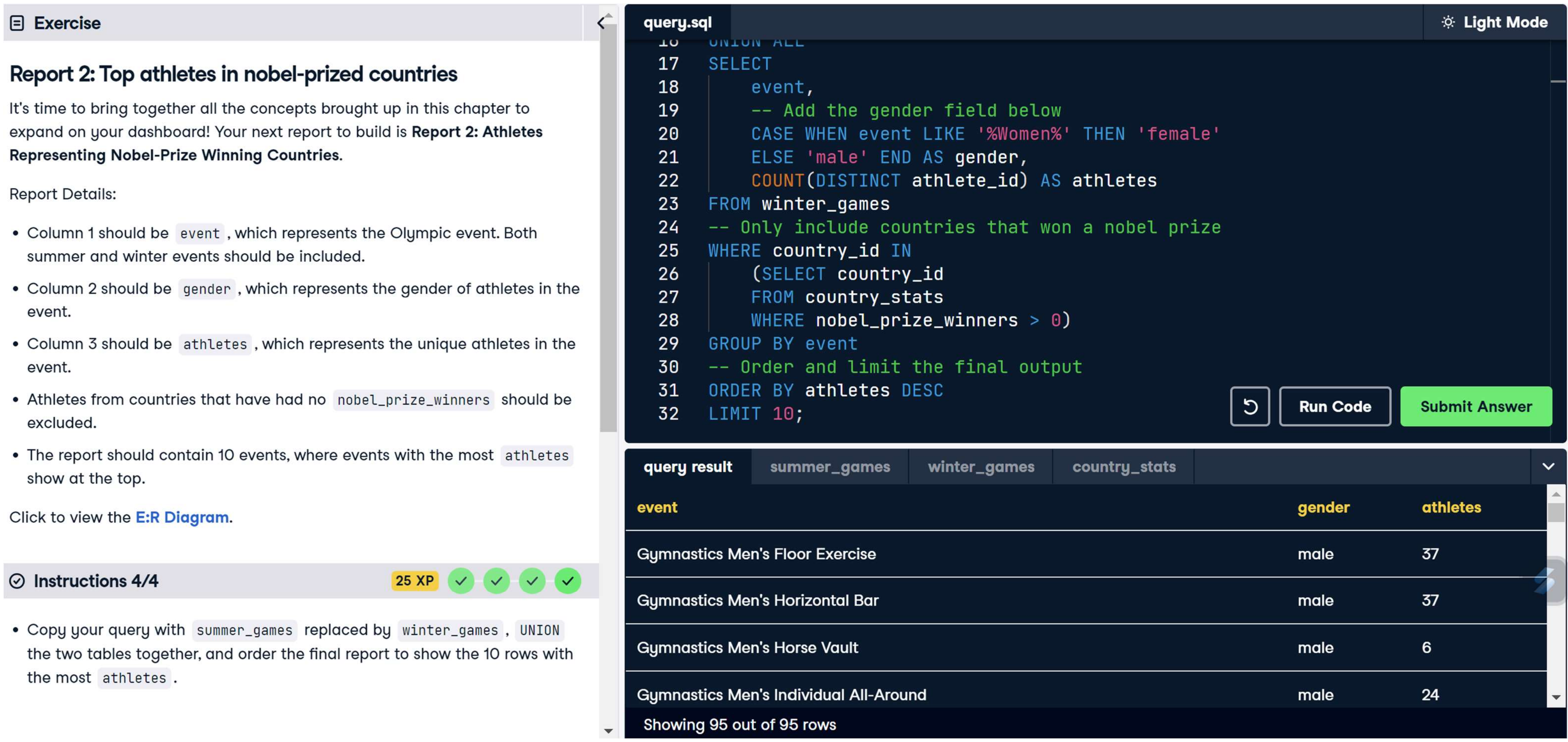
Instructions 4/4 25 XP

- Copy your query with `summer_games` replaced by `winter_games`, `UNION` the two tables together, and order the final report to show the 10 rows with the most `athletes`.

query.sql

```sql
-- Pull event and unique athletes from summer_games
SELECT
    event,
    -- Add the gender field below
    CASE WHEN event LIKE '%Women%' THEN 'female'
    ELSE 'male' END AS gender,
    COUNT(DISTINCT athlete_id) AS athletes
FROM summer_games
-- Only include countries that won a nobel prize
WHERE country_id IN
    (SELECT country_id
    FROM country_stats
    WHERE nobel_prize_winners > 0)
GROUP BY event
-- Add the second query below and combine with a UNION
UNION ALL
SELECT
```

```
event                                   gender    athletes
Gymnastics Men's Floor Exercise         male      37
Gymnastics Men's Horizontal Bar         male      37
Gymnastics Men's Horse Vault            male      6
Gymnastics Men's Individual All-Around  male      24
Showing 95 out of 95 rows
```



Exercise

Report 2: Top athletes in nobel-prized countries

It's time to bring together all the concepts brought up in this chapter to expand on your dashboard! Your next report to build is **Report 2: Athletes Representing Nobel-Prize Winning Countries.**

Report Details:

- Column 1 should be `event`, which represents the Olympic event. Both summer and winter events should be included.
- Column 2 should be `gender`, which represents the gender of athletes in the event.
- Column 3 should be `athletes`, which represents the unique athletes in the event.
- Athletes from countries that have had no `nobel_prize_winners` should be excluded.
- The report should contain 10 events, where events with the most `athletes` show at the top.

Click to view the E:R Diagram.

Instructions 4/4 25 XP

- Copy your query with `summer_games` replaced by `winter_games`, `UNION` the two tables together, and order the final report to show the 10 rows with the most `athletes`.

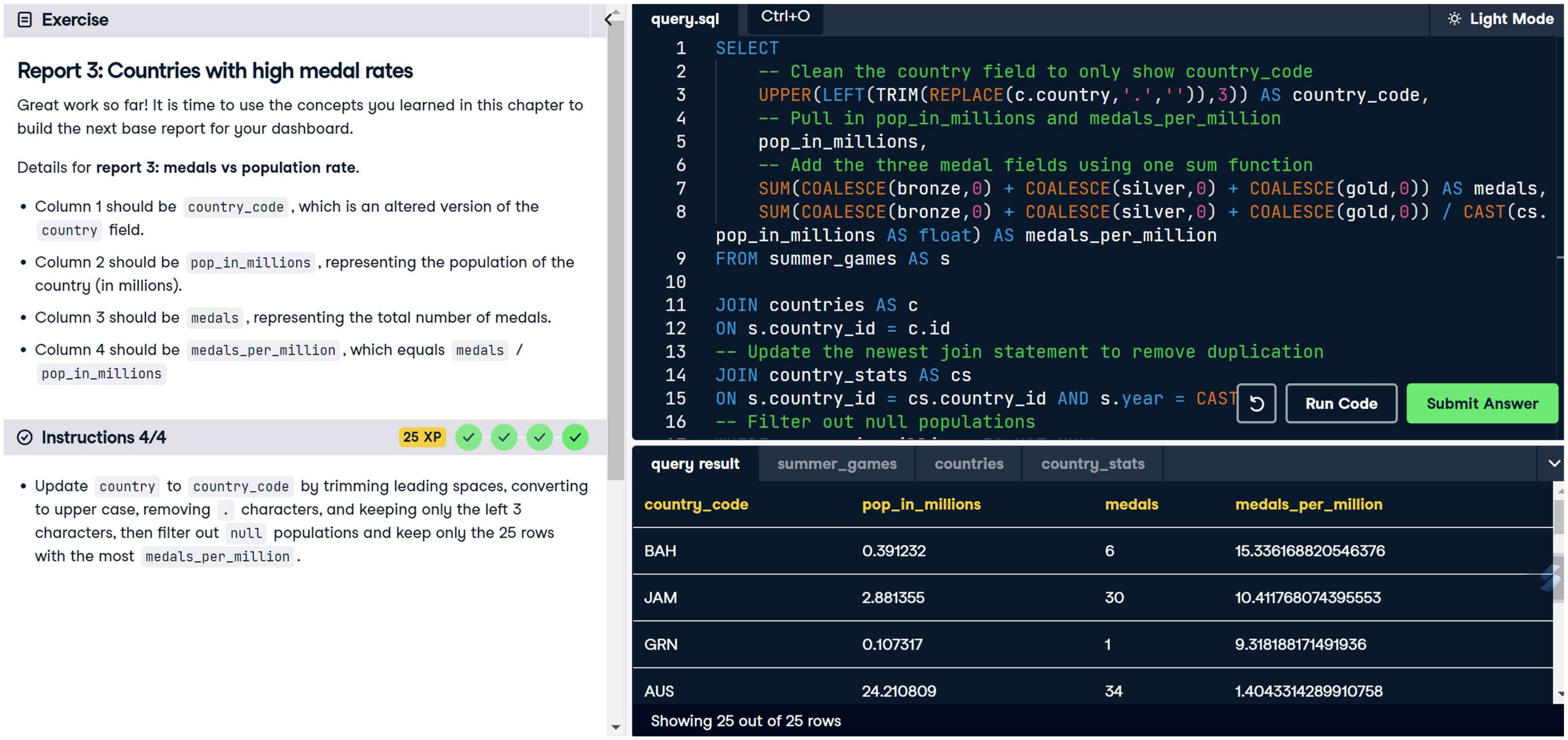
query.sql

```sql
UNION ALL
SELECT
    event,
    -- Add the gender field below
    CASE WHEN event LIKE '%Women%' THEN 'female'
    ELSE 'male' END AS gender,
    COUNT(DISTINCT athlete_id) AS athletes
FROM winter_games
-- Only include countries that won a nobel prize
WHERE country_id IN
    (SELECT country_id
    FROM country_stats
    WHERE nobel_prize_winners > 0)
GROUP BY event
-- Order and limit the final output
ORDER BY athletes DESC
LIMIT 10;
```

query result | summer_games | winter_games | country_stats

event	gender	athletes
Gymnastics Men's Floor Exercise	male	37
Gymnastics Men's Horizontal Bar	male	37
Gymnastics Men's Horse Vault	male	6
Gymnastics Men's Individual All-Around	male	24

Showing 95 out of 95 rows



Report 3: Countries with high medal rates

Great work so far! It is time to use the concepts you learned in this chapter to build the next base report for your dashboard.

Details for **report 3: medals vs population rate**.

- Column 1 should be `country_code`, which is an altered version of the `country` field.
- Column 2 should be `pop_in_millions`, representing the population of the country (in millions).
- Column 3 should be `medals`, representing the total number of medals.
- Column 4 should be `medals_per_million`, which equals `medals` / `pop_in_millions`

Instructions 4/4

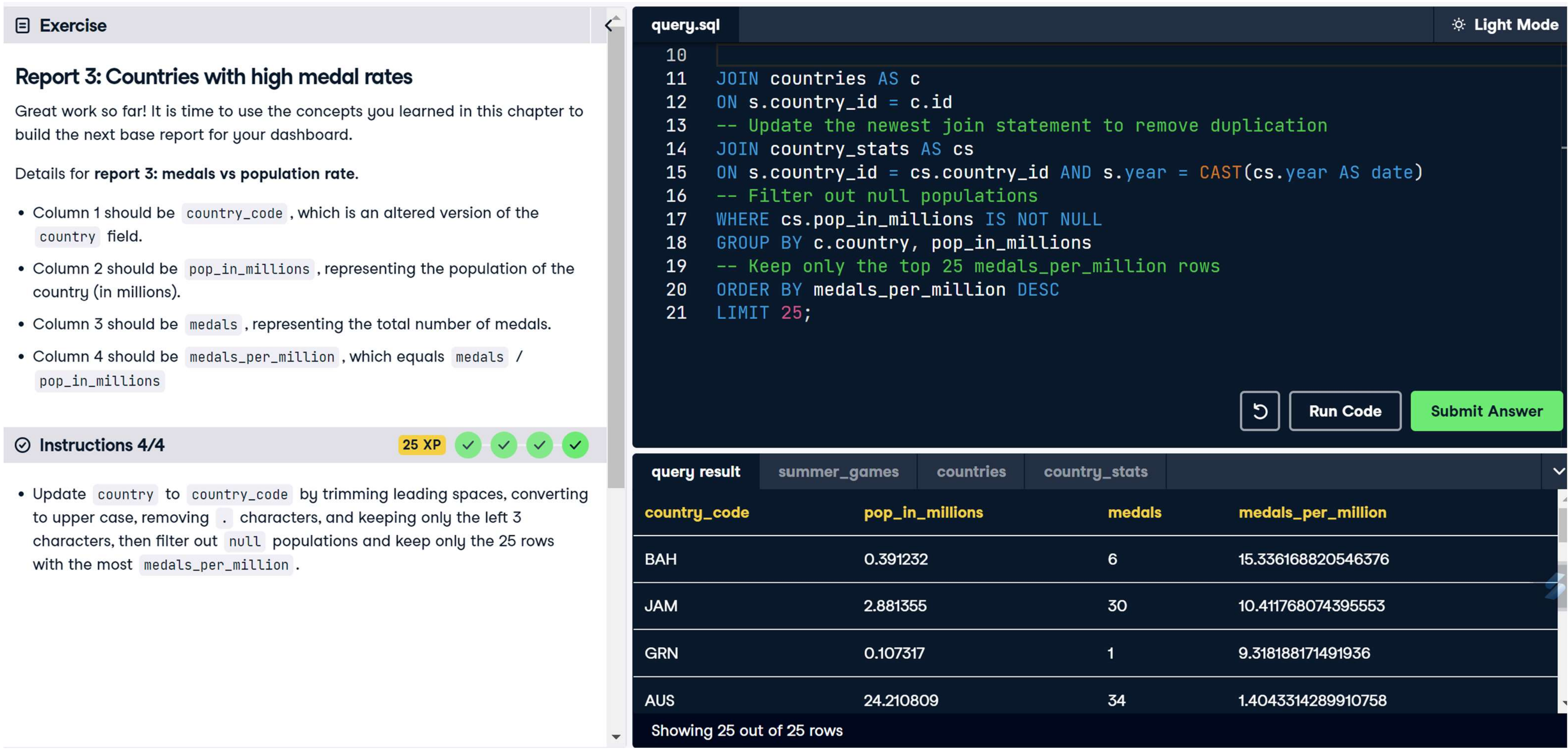
- Update `country` to `country_code` by trimming leading spaces, converting to upper case, removing `.` characters, and keeping only the left 3 characters, then filter out `null` populations and keep only the 25 rows with the most `medals_per_million`.

```sql
SELECT
    -- Clean the country field to only show country_code
    UPPER(LEFT(TRIM(REPLACE(c.country,'.','')),3)) AS country_code,
    -- Pull in pop_in_millions and medals_per_million
    pop_in_millions,
    -- Add the three medal fields using one sum function
    SUM(COALESCE(bronze,0) + COALESCE(silver,0) + COALESCE(gold,0)) AS medals,
    SUM(COALESCE(bronze,0) + COALESCE(silver,0) + COALESCE(gold,0)) / CAST(cs.
pop_in_millions AS float) AS medals_per_million
FROM summer_games AS s

JOIN countries AS c
ON s.country_id = c.id
-- Update the newest join statement to remove duplication
JOIN country_stats AS cs
ON s.country_id = cs.country_id AND s.year = CAST
-- Filter out null populations
```

country_code	pop_in_millions	medals	medals_per_million
BAH	0.391232	6	15.336168820546376
JAM	2.881355	30	10.411768074395553
GRN	0.107317	1	9.318188171491936
AUS	24.210809	34	1.4043314289910758

Showing 25 out of 25 rows



Report 3: Countries with high medal rates

Great work so far! It is time to use the concepts you learned in this chapter to build the next base report for your dashboard.

Details for **report 3: medals vs population rate**.

- Column 1 should be `country_code`, which is an altered version of the `country` field.
- Column 2 should be `pop_in_millions`, representing the population of the country (in millions).
- Column 3 should be `medals`, representing the total number of medals.
- Column 4 should be `medals_per_million`, which equals `medals` / `pop_in_millions`

- Update `country` to `country_code` by trimming leading spaces, converting to upper case, removing `.` characters, and keeping only the left 3 characters, then filter out `null` populations and keep only the 25 rows with the most `medals_per_million`.

```sql

JOIN countries AS c
ON s.country_id = c.id
-- Update the newest join statement to remove duplication
JOIN country_stats AS cs
ON s.country_id = cs.country_id AND s.year = CAST(cs.year AS date)
-- Filter out null populations
WHERE cs.pop_in_millions IS NOT NULL
GROUP BY c.country, pop_in_millions
-- Keep only the top 25 medals_per_million rows
ORDER BY medals_per_million DESC
LIMIT 25;
```

country_code	pop_in_millions	medals	medals_per_million
BAH	0.391232	6	15.336168820546376
JAM	2.881355	30	10.411768074395553
GRN	0.107317	1	9.318188171491936
AUS	24.210809	34	1.4043314289910758

Showing 25 out of 25 rows